

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently amended) A method comprising:
  - a ~~content server~~ receiving at a content server information from at least one content provider;
    - sending at least one portion of the information to a user terminal for display on the user terminal;
    - receiving notification of active keys, the active keys associated with the at least one portion of the information displayed on the user terminal;
    - the content server determining at the content server if any of the at least one portion of the information has changed by identifying one or more differences between the at least one portion of the information and prior information previously stored in a data store of the content server;
    - updating in the data store the information from the at least one content provider that has changed; and
    - transmitting to the user terminal the information associated with the active keys from the at least one content provider that has changed without also transmitting unchanged information, the changed information being real-time information.
2. (Original) The method recited in claim 1, wherein the information comprises a plurality of real-time data values from the content provider.
3. (Previously presented) The method recited in claim 2, wherein the updating of information from the content provider further comprises:
  - accessing a hash table containing a plurality of prior real-time data values using a plurality of keys associated with the plurality of real-time data values;
  - determining whether the plurality of real-time data values received from the content provider has changed from the prior plurality of real-time data values contained in the hash table; and

updating the prior plurality real-time data values contained in the hash table with the plurality of real-time values received from the content provider when the plurality of real-time data values received from the content provider has changed from the plurality of prior real-time data values contained in the hash table.

4. (Currently amended) The method recited in claim 3, wherein the transmitting of the plurality of real-time data values associated with the active keys that have been updated in the hash table to the user terminal further comprises:

activating a data thread when a real-time data value of the plurality of prior real-time data values is updated in the hash table;

determining the position on a screen of the user terminal corresponding to the real-time data value;

transmitting the real-time data value associated with an active key to the user terminal for display on the screen of the user terminal in the position indicated.

5. (Previously Presented) The method recited in claim 4, wherein the activating step comprises activating the data thread using remote method invocation.

6. (Previously presented) The method recited in claim 3, further comprising:  
spawning a data server thread in response to receiving a connection request from the user terminal;  
retrieving, by the data thread, a user defined portfolio containing a plurality of keys;

generating an activated HTML page containing an embedded applet and sending the activated HTML page to the user terminal;

monitoring the plurality of keys contained in the user defined portfolio; and  
identifying currently active keys of said of the plurality of keys.

7. (Cancelled)

8. (Currently amended) The method recited in claim 6, comprising:  
determining whether a shutdown request was made; and  
disconnecting all connections to the user terminal when the shutdown request was made.

9. (Original) The method recited in claim 8, comprising:  
retrieving the plurality of real-time data values on a periodic basis.
10. (Previously Presented) The method recited in claim 9, comprising:  
notifying a data server thread when a real-time data value of the plurality of data that values have changed.
11. (Previously presented) The method recited in claim 6, comprising:  
determining whether a page changed is required;  
receiving, by the data server thread, a plurality of new active keys; and  
transmitting the plurality of real-time data values to the user terminal through the data server thread.
12. (Currently amended) A computer program executable by computer and embodied on a computer readable medium comprising:  
a real-time data server code segment to receive real-time data values from at least one content provider, receive active keys that are associated with the real-time data values from at least one user terminal, determine if any of the real-time data values have changed from a prior real-time data values by identifying one or more differences between the real-time data values and the prior real-time data values, and transmit the changed real-time data values associated with the active keys without also transmitting unchanged data values to the at least one user terminal when any of the real-time data values associated with active keys have changed from the prior real-time data values.
13. (Original) The computer program recited in claim 12, wherein the real-time data server code segment further comprises:  
a hash table storing the prior real-time data values and being updated when the real-time data values from the content provider have changed from the prior real-time data values.
14. (Previously Presented) The computer program recited in claim 13, wherein the real-time data server code segment further comprises:  
a web engine server module code segment to access a database having a portfolio generated by a user and generate an HTML page and applet, wherein upon receipt of a

connection request from the user terminal the web engine server module code segment downloads the HTML page and applet to the user terminal code segment.

15. (Previously Presented) The computer program recited in claim 13, wherein the real-time data server code segment further comprises:

a source filter server module code segment to receive real-time data values from a content provider and determine if the real-time data values have changed from prior real-time data values stored, and activate a data thread code segment when the real-time data values have changed from prior real-time data values.

16. (Previously Presented) The computer program recited in claim 15, wherein the real-time data server code segment further comprises:

a realtime data server module code segment to communicate between the user terminal code segment and the source filter server module code segment through the data server thread code segment.

17. (Previously Presented) The computer program recited in claim 16, further comprising:

a source filter server module code segment to receive the real-time data values from the content provider; and update the hash table.

18. (Cancelled).

19. (Previously presented) The computer program recited in claim 13, further comprising:

a web server module code segment to communicate to the user terminal code segment and retrieve a portfolio specified by the user terminal code segment from a database; and

a pagination engine module code segment, in communication with the web server module code segment, to create the HTML page and applet code segment based on the portfolio selected and the size of a screen on a user terminal.

20. (Currently amended) A system comprising:

a real-time data server configured to receive real-time data values from at least one content provider, receive active keys that are associated with the real-time data values from at least one user terminal, determine if any of the real-time data values have changed from prior real-time data values by identifying one or more differences between the real-time data values and the prior real-time data values, and transmit the changed real-time data values associated with the active keys without also transmitting unchanged data values to the at least one user terminal when any of the real-time data values associated with the active keys have changed from the prior real-time data values, each at least one user terminal capable of displaying different said real-time data values associated with the active keys from different said at least one content providers simultaneously.

21. (Original) The system recited in claim 20, wherein the real-time data server further comprises:

a hash table storing the prior real-time data values and being updated when the real-time data values from the content provider have changed from the prior real-time data values.

22. (Previously presented) The system recited in claim 21, wherein the real-time data server further comprises:

a web engine server module to access a database having a portfolio generated by a user and generate an HTML page and applet, wherein upon receipt of a connection request from the user terminal the web engine server module transmits the HTML page and applet to the user terminal.

23. (Original) The system recited in claim 21, wherein the real-time data server further comprises:

a source filter server module to receive real-time data values from the content provider and determine if the real-time data values have changed from prior real-time data values stored in table, and activate a data thread when the real-time data values have changed from prior real-time data values.

24. (Original) The system recited in claim 23, wherein the real-time data server further comprises:

a realtime data server module to communicate between the user terminal and the source filter server module through the data server thread.

25. (Previously Presented) The system recited in claim 24, further comprising:  
a source filter server module to receive the real-time data values from the content provider; and update hash table.

26. (Cancelled).

27. (Original) The computer program recited in claim 22, wherein the web engine server module further comprises:

a web server module to communicate to the user terminal and retrieve a portfolio specified by the user terminal from a database; and

a pagination engine module, in communication with the web server module, to create HTML page and applet based on the portfolio selected and the size of the screen on the user terminal.

28. (Currently amended) A network node comprising:

a transmitting interface for transmitting information from a content provider to a user terminal, said information containing at least a portion to be displayed on the user terminal;

a monitoring interface for monitoring additional information from the content provider to determine if any of the portion of the information transmitted to the user terminal has changed, wherein the determination is made by identifying one or more differences between the portion of the information transmitted to the user terminal and the additional information; and

an update module for updating the information from the content provider that has changed,

wherein the transmitting interface transmits the information from the content provider that has changed and is displayed on the user terminal to the user terminal without also transmitting unchanged information.

29. (Previously Presented) The node according to claim 28, wherein the information comprises a plurality of real-time data values from the content provider.

30. (Previously Presented) The node according to claim 28, wherein the updating of information from the content provider further comprises:

accessing a hash table containing a plurality of prior real-time data values using a plurality of keys associated with the plurality of real-time data values;

determining whether the plurality of real-time data values received from content provider has changed from the prior plurality of real-time data values contained in the hash table; and

updating the prior plurality real-time data values contained in the hash table with the plurality of real-time values received from the content provider when the plurality of real-time data values received from content provider has changed from the plurality of prior real-time data values contained in the hash table.

31. (Previously Presented) The node according to claim 30, wherein the network node retrieves a user defined portfolio by a data thread containing a plurality of keys, the data thread being received from the user terminal, generates activated HTML page containing an embedded applet and downloads the activated HTML page to the user terminal.

32. (Previously Presented) The node according to claim 30, wherein the network node monitors the plurality of keys contained in the user defined portfolio and identifies currently active keys of said plurality of keys.

33. (Previously Presented) The node according to claim 32, wherein the network node reads the currently active keys, determines if the currently active keys have changed, updates the hash table with the real-time data values for currently active keys, and downloads the real-time values for the currently active keys that have changed from the hash table to the user terminal.

34. (Previously Presented) The node according to the claim 33, wherein the network node determines whether a shutdown request was made and disconnects all connections to the user terminal when the shutdown request was made.

35. (Currently amended) A real-time server computer comprising memory storing computer executable code modules that each comprise computer executable instructions stored in the memory, said code modules comprising:

    a source filter server module that receives data from a real-time content provider, and stores the received data in a keyed hash table;

    a real-time data server module comprising submodules including:

        a client connection submodule that establishes a data server thread connection with a remote mobile terminal;

        wherein when any data server thread connection receives an active key request from the remote mobile terminal, the real-time data server module performs a method including a) querying the keyed hash table for corresponding data; b) determining whether the queried data differs from data previously sent to the remote mobile terminal; and c) when the queried data differs from the data previously sent to the remote mobile terminal, the queried data is sent to the remote mobile terminal, otherwise the queried data is not sent to the remote mobile terminal; and

        a web engine server module that communicates formatted data to the remote mobile terminal based on the queried data.